NASA Technical Memorandum 100521

ESTIMATING THE DISTRIBUTION OF FAULT LATENCY IN A DIGITAL PROCESSOR

(NASA-TM-100521) ESTIMATING THE DISTRIBUTION OF FAULT LATENCY IN A DIGITAL PROCESSOR (NASA) 22 p CSCL 09B

N88-14666

Unclas

G3/62 0114262

Erik L. Ellis and Ricky W. Butler

NOVEMBER 1987

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

INTRODUCTION

In this paper a new technique is presented for estimating fault latency in a digital processor. Fault latency is defined as the time interval from the occurrence of a fault until the generation of an error. The time interval from the error generation until the fault is detected will be referred to as error latency. Unfortunately, the time of error generation is not directly observable. Therefore, traditionally the sum of fault-latency and error latency has been measured [McGough, 1981, McGough 1983]. Chillarege and Iyer have presented a method of measuring fault latency in memory [Chillarege and Iyer, 1986]. Since this method relies on periodic memory dumping to observe the approximate time of error generation, the method does not apply to other sections of a digital processor. Recently Shin has presented a general method of measuring the fault-latency of injected faults in a physical processor [Shin, 1986]. This paper improves Shin's methodology by providing a more powerful statistical method to analyze the results of experimentation.

Fault latency is important for several reasons. First, as long as a fault lies latent in a processor, the fault cannot be detected, and, therefore, it cannot be removed. In a fault-tolerant system the longer a fault lies latent, the greater the probability that a fault will arrive in a second processor, and, at some later time, both faults will produce errors simultaneously. These errors could defeat the voter and the system would fail. For these reasons, some reliability analysis tools require estimates of certain characteristics of fault latency [Geist, 1983; Bavuso 1985]. Second, if error propagation times in separate channels are not independent, long latency times can be particularly disastrous. The likelihood of simultaneous error manifestations can be significantly larger than in the independent case. An effective method for measuring fault latency is clearly a prerequisite to investigations into the nature of error correlations.

In Shin's study, an indirect method of measuring the fault latency distribution was described. One drawback to this technique was noted by the authors -- the technique could yield an empirical distribution with decreasing intervals. This occurs because the points on the distribution are estimated independently of each other.

In this paper statistical methods are presented which provide a non-decreasing estimate of the distribution function and confidence bounds for points on the distribution function. The details of the experimental process

are described along with the results of an experiment which was performed on the SIFT computer system.

THE EXPERIMENTAL METHOD

Since fault latency in a digital processor is not deterministic, it is necessary to model it with a random variable L_f. If the latency time were directly measurable, a histogram of fault latency times could be constructed from which characteristics of the distribution function could be inferred. Unfortunately, this cannot be done. Therefore, an indirect method was proposed by Shin [Shin, 1986].

Let F(t) represent the distribution function of fault latency:

$$F(t) = \Pr\{L_f \leq t\}. \tag{1}$$

Next, consider the injection of a fault of duration t_i.[1] The moment of error generation is the end point of the fault latency period (see fig. 1).

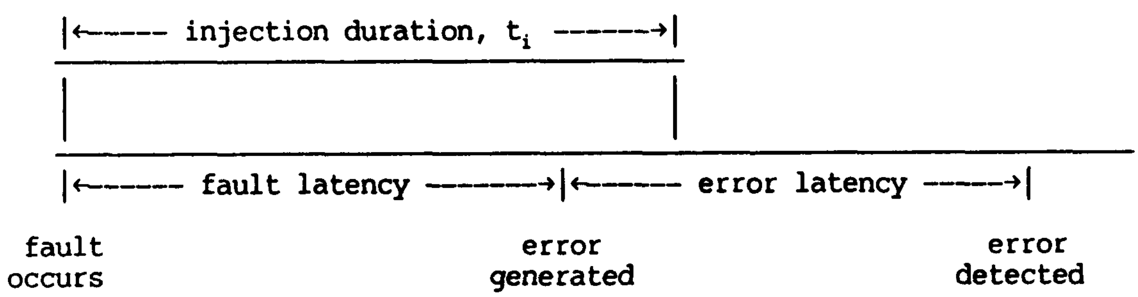

Figure 1. - Fault Latency/Error Latency Periods

1 It is necessary to artificially introduce faults into the digital processor for two reasons: (1) natural faults occur too infrequently to obtain statistically significant quantities of data. (2) the time of occurrence of a natural fault is not measurable.

$E_g^{(i)}$ = "An error is generated during a fault injection of duration t_i"

$E_d^{(i)}$ = "An error is detected during/after an injection of duration t_i"

Since the moment of error generation can only occur in the presence of a fault, the event $E_g^{(i)}$ is equivalent to the event "$L_f \le t_i$." Under the assumption that all errors generated are eventually detected, the event $E_d^{(i)}$ is equivalent to the event $E_g^{(i)}$. Let J_i be an indicator variable for the event $E_d^{(i)}$. Then

$$
\begin{aligned}
F(t_i) &= \Pr\{L_f \le t_i\} && (2)\\
&= \Pr\{E_d^{(i)}\} && (3)\\
&= \Pr\{J_i = 1\}. && (4)
\end{aligned}
$$

Therefore, the random variable J_i has a Bernoulli distribution with parameter $F(t_i)$.

METHODOLOGY FOR ESTIMATION OF FAULT LATENCY DISTRIBUTION

The method for estimating $F(t)$ at $t = t_i$: $i = 1,\ldots r$ is a five step process:

(1) sample fault injection locations proportional to location failure rates

To characterize the average behavior of fault latency in a processor, data is gathered from faults injected at various locations in the processor. These locations are chosen proportional to their failure rates. In other words, if there are M possible injection locations in the processor with known failure rates $z_i,\ldots,z_M$ then location j is sampled with probability $z_j/(\Sigma_1^M z_i)$.

(2) inject n_i faults of duration t_i at each of the sampled locations: $i = 1,\ldots r$

After each injection, record whether an error is detected or not. Since the analysis method presented in this paper does not take censoring into account, it is assumed that the error detection mechanism is perfect. In other

words, a fault is injected and a period of time C elapses. At the end of that elapsed period if no errors have been detected it is assumed that the fault injection has produced no errors and will produce no errors in the future (i.e., $L_f = \infty$). The choice of an appropriate C may be restricted by some maximum 'waiting time' allowed by the experimental system. If not, C may be chosen to be some value suitably greater than any error detection times observed in previous fault injection experiments.

(3) let D_i = "the number of injections in which the event $E_d^{(i)}$ occurred": $i = 1, \ldots r$

(4) let $\hat{F}(t_i) = D_i/n_i$: $i = 1, \ldots r$

Since J_i has the Bernoulli distribution with parameter $F(t_i)$ and the Bernoulli trials are independent, D_i is binomially distributed with parameters $F(t_i)$ and n_i. The maximum likelihood estimate (MLE) and uniform minimum variance unbiased estimate (UMVUE) of $F(t_i)$ is $\hat{F}(t_i) = D_i/n_i$. $\hat{F}(t_i)$ is, in fact, the Shin estimate of $F(t_i)$.

The problem with the Shin estimate is that while the distribution function, by definition, is non-decreasing, the Shin estimates may not be. That is, it is possible to observe:

$$\hat{F}(t_i) > \hat{F}(t_j), \text{ for some } i,j: \ 1 \leq i < j \leq r \qquad (5)$$

Therefore a fifth step is proposed.

(5) apply isotonic regression to the points $\{\hat{F}(t_i)\}_{i=1}^{r}$ with respective weights $\{n_i\}_{i=1}^{r}$

Isotonic regression enables one to take into account the monotonic property of the distribution in the data analysis. It has been shown that isotonic regression is the maximum likelihood estimate and the least-squares estimate of the sequence $F(t_1, \ldots, F(t_r)$ over the domain of non-decreasing sequences [Barlow, et. al., 1972]. Consequently, the isotonic regression estimate is always non-decreasing. Since, theoretically, the fault latency distribution must be non-decreasing, the use of isotonic regression is justified.

The method presented here to compute the isotonic regression is based on

the "greatest convex minorant" graphical interpretation given in [Barlow, et. al.,1972].

(a) $G_i = \Sigma_1^i \hat{F}(t_j)n_j$: $i = 1,\ldots,r$
(b) $W_i = \Sigma_1^i n_j$: $i = 1,\ldots,r$
(c) $G_0 = W_0 = 0$
(d) $S_{i,j} = (G_j - G_i)/(W_j - W_i)$: $i = 0,\ldots,r-1$ and $j = i+1,\ldots,r$
(e) $M_i = \min\{S_{i,i+1},\ldots,S_{i,r}\}$: $i = 0,\ldots,r-1$
(f) $m(i) = \max\{j: S_{i,j} = M_i\}$: $i = 0,\ldots,r-1$
(g) Then $F^*(t_i) = M_{m(0)}$: $i = 1,\ldots,m(0)$
    and $F^*(t_i) = M_{m(m(0))}$: $i = m(0)+1,\ldots,m(m(0))$
    and $F^*(t_i) = M_{m(m(m(0)))}$: $i = m(m(0))+1,\ldots,m(m(m(0)))$
    and so on until $F^*(t_r)$ is computed.

The resulting sequence, $F^*(t_1),\ldots,F^*(t_r)$, is the isotonic regression of $\hat{F}(t_1),\ldots,\hat{F}(t_r)$ with weights $n_1,\ldots,n_r$.

METHODOLOGY FOR COMPUTING CONFIDENCE BANDS FOR THE FAULT LATENCY DISTRIBUTION

While very little has been published on the subject of computing confidence intervals under order restrictions, a technique was developed recently by David A. Schoenfeld to compute isotonic confidence bounds for a sequence of normal ordered means [Schoenfeld, 1986]. To compute a $1-\alpha$ confidence interval, the isotonic confidence bounds method will be employed to find the upper $1-\alpha/2$ isotonic confidence bound and then the lower $1-\alpha/2$ isotonic confidence bound.

The upper isotonic confidence bounds, FU_i, for $F(t_i)$ is computed by testing the appropriate hypothesis. The basic idea is to let FU_i be the upper bound of the acceptance region of this test using the likelihood ratio test statistic. The likelihood ratio test statistic is a function of the MLE. In the isotonic confidence bounds method, however, isotonic regression estimates (which are MLE's over the domain of non-decreasing sequences) are used instead. In this manner, the non-decreasing property of the distribution is taken into account. The method itself may be described in five steps.

(1) compute the estimate $\hat{F}(t_i) = D_i/n_i$ for $i = 1,\ldots,r$

(2) perform a variance stabilizing transformation on the Shin estimates

Assume $\hat{F}(t_1),\ldots,\hat{F}(t_r)$ are statistically independent. Let

$$Y_i = \text{Arcsin}\{[\hat{F}(t_i)]^{1/2}\},\ i = 1,\ldots,r \tag{6}$$

Then $Y_1,\ldots,Y_r$ are independent and, if n_i is reasonably large, Y_i is approximately normally distributed with unknown mean u_i and known variance $(4n_i)^{-1}$: $i = 1,\ldots,r$. The problem now is to find upper isotonic confidence bounds for $u_1,\ldots,u_r$. Therefore the appropriate hypothesis tests mentioned above are $H_0: x \leq u_i$: $i = 1,\ldots,r$.

(3) compute critical value c_i: $i = 1,\ldots,r$

The critical values of the tests are the solutions, $c_1,\ldots,c_r$, to the equations

$$\alpha/2 = \Sigma_{j=1}^{r-i+1}\ g(j,r-i+1)[1-\chi_j(c_i)],\ i = 1,\ldots,r \tag{7}$$

where g(p,q) is the probability that the isotonic regression estimates of q normally distributed random variables with mean zero and variance n_i^{-1}: $i = 1,\ldots,q$, will have C distinct negative values. This g function is easily approximated by a Monte Carlo simulation. χ_j is the cumulative distribution function of the Chi-squared distribution with j degrees of freedom.

(4) compute ν_i, the upper isotonic confidence bound for u_i: $i = 1,\ldots,r$

The ν_i's are the solutions to the equations

$$T_i(\nu_i) = c_i,\ i = 1,\ldots,r \tag{8}$$

Where $T_i(x)$ is defined as

$$T_i(x) = \sigma^{-2}\{\Sigma_A\ n_j(x-u_j^*)^2 - \Sigma_B\ n_j(x-\tilde{u}_j)^2\} \tag{9}$$

and $u_1^*,\ldots,u_r^*$ are the isotonic regression estimates of $Y_1,\ldots,Y_r$; $\tilde{u}_1,\ldots,\tilde{u}_{i-1}$ are the isotonic regression estimates of $Y_1,\ldots,Y_{i-1}$; Σ_A denotes summation

over the set $\{j: u_i^* \leq u_j^* \leq x\}$; and Σ_B denotes summation over the set $\{j: j < i, u_i^* \leq \tilde{u}_j \leq x\}$.

(5) perform the inverse variance stabilizing transformation on, ν_i, the isotonic confidence bound for u_i: $i = 1,\ldots,r$

The upper bounds, $FU_1,\ldots,FU_r$, are computed by performing the inverse of the variance stabilizing transformation performed earlier.

$$FU_i = \sin^2(\nu_i),\ i = 1,\ldots,r \tag{10}$$

The $1-\alpha/2$ lower bounds, $FL_1,\ldots,FL_r$, are computed by inverting the order of the n_i's and inverting the order and sign of the Y_i's, and then solving for the upper $1-\alpha/2$ bounds as above. That is

$$Y_i' = -Y_{r-i+1},\ i = 1,\ldots,r \tag{11}$$

$$n_i' = n_{r-i+1},\ i = 1,\ldots,r \tag{12}$$

Then perform steps 3 and 4 above to obtain $\nu_1',\ldots,\nu_r'$. To transform the $\nu_1',\ldots,\nu_r'$ into lower isotonic confidence bounds on $F(t_1),\ldots,F(t_r)$, the sign and order must also be restored. Therefore

$$FL_i = -\sin^2(\nu_{r-i+1}'),\ i = 1,\ldots,r \tag{13}$$

The pairs (FL_i, FU_i) define a $1-\alpha$ isotonic confidence interval for the distribution value $F(t_i)$.

Since the FL_i's and FU_i's are random variables, it is only approximately true that

$$\Pr\{FL_i \leq F(t_i) \leq FU_i\} = 1-\alpha,\ i = 1,\ldots,r \tag{14}$$

It is not necessarily true, however, that

$$\Pr\{[FL_1 \leq F(t_1) \leq FU_1] \text{ and, } \ldots, \text{ and } [FL_r \leq F(t_r) \leq FU_r]\} = 1-\alpha \tag{15}$$

The probability of this event is normally somewhat less than $1-\alpha$.

SIFT: THE EXPERIMENTAL SYSTEM

Software Implemented Fault Tolerance (SIFT) is an experimental fault tolerant computer system developed for NASA Langley Research Center as a testbed for fault tolerant systems research. [Goldberg, et. al., 1984] The system consists of up to six BDX 930 processors. The processors communicate via a fully-connected communications network. The processors are frame-synchronized by the interactive-convergence clock synchronization algorithm. The system obtains fault-tolerance through the use of task replication and exact-match voting of the outputs of identical tasks. The tasks are scheduled according to a static schedule table which is performed cyclically. Tasks are dispatched in response to clock interrupts. The following time-graph illustrates this process:

```
--------|------------------------------------|-----
        | vote    |   task execution         |
      clock                                clock
    interrupt                            interrupt
```

Since the results of identical tasks are voted in an exact-match manner all propagated errors are eventually detected. There is, however, no guaranteed bound on the error latency. Also, while the faults are latent, they are not detectable. The experimental approach discussed in this paper for measuring fault latency depends critically upon the existence of such a voting technique which can detect 100% of all propagated errors.

The SIFT operating system was instrumented to obtain the time of each error detection on the non-injected processors. This time was obtained on each processor from a global clock with millisecond resolution. Since error detection is accomplished by voting, error detection is possible only in subframes where voting occurs. A board is attached to an extender board, and the fault injector clip is physically connected to the chip which is to be faulted.

A fault may or may not generate errors which are detectable by the operating system's voters. Figure 2 illustrates the the effect of an injected fault:

```
———————————————————————————————— ... ——————————————————> t
    ↑          ↑  ↑  ↑    ↑    ↑  ↑          ↑    ↑    ↑
    s          e1 e2 e3   e4   e5 e6  ...    en   r    C
```

where

s = time fault injection initiated

e_i = the time of detection of the ith error $(1 \leq i \leq n)$

r = time operating system reconfigures (if reconfiguration occurs)

C = censoring point (i.e. point where experimental observation is terminated)

Figure 2. - Effect of an injected fault

The methodology for estimating the distribution of fault latency does not require the error detections times. The only information needed is whether or not an error is detected.

It is impractical to perform fault injections at every pin in a processor. Thus, the fault injection locations were chosen randomly weighted according to the chip failure rates and a small set of fault durations (to be injected at every randomly-selected pin) were predetermined. The chip failure rates were determined using MIL-STD-217D. The injection locations (25) were sampled proportional to the rate of chip failure. There were six locations in SIFT where injections (eg., power supply pins, ground pins) could not be performed and were removed from the sample. The remaining 19 locations constituted the sample of fault injection locations. These locations are enumerated in table 1.

Board	Chip Type	Chip Number	Pin
CPU	AM2901A	U38	21
CPU	AM2901A	U35	26
CPU	AM2901A	U29	21
CPU	54S151	U4	2
CPU	54S288	U70	13
MPM1	MK4114.3	U21	11
MPM1	MK4114.3	U12	12
MPM1	MK4114.3	U45	14
MPM1	54LS04	U59	12
MPM2	MK4114.3	U21	16
MPM2	MK4114.3	U11	15
MPM2	MK4114.3	U17	4
MPM2	MK4114.3	U41	8
MPM2	MK4114.3	U52	6
MPM2	MK4114.3	U37	6
MPM2	MK4114.3	U35	12
MPM2	MK4114.3	U28	5
MPM2	54LS04	U53	1
BR	54LS244	U81	11

Table 1 - Fault Injection Locations

The set of fault durations were not chosen to be equally far apart (i.e. equal successive differences). This is impractical since the fault latency is several orders of magnitude longer for some pins than for others. A spacing appropriate for one pin location in the processor would not be appropriate for another. Ten injection durations (i.e., $t_i : i = 1,...,r$) were chosen log-linearly from 0 to 64 seconds. After injecting at each location for these durations, a preliminary data analysis revealed that more detail in the shape of the distribution was required for some intervals on the abscissa. Therefore, additional t_i's were chosen between the previously chosen t_i's. The full list of injection durations are listed in Table 2.

0.001	0.002	0.003	0.005	0.008	0.010	0.017
0.024	0.032	0.054	0.077	0.100	0.208	0.316
0.554	0.772	1.000	2.081	3.162	5.442	7.721
10.00	20.81	31.62	100.000	316.220	1000.000	

Table 2. - Fault Injection Durations (ms)

Five stuck at logical one and five stuck at logical zero faults were performed at each pin location of table 1 for each of the fault durations listed in table 2. A total of 4860 faults were injected — 180 injections for each t_i.

RESULTS AND ANALYSIS OF SIFT DATA

The isotonic regression estimate of the fault latency distribution function and the 95% isotonic confidence bounds are given in table 3 and plotted in figure 3.

t_i (ms)	$F^*(t_i)$	95% Confidence Bounds
0.001	0.11842	(0.07852, 0.16571)
0.002	0.11842	(0.08439, 0.17784)
0.003	0.13684	(0.09508, 0.21414)
0.005	0.20000	(0.13474, 0.25968)
0.008	0.20526	(0.15403, 0.27763)
0.010	0.22632	(0.17031, 0.31761)
0.017	0.31053	(0.22813, 0.39647)
0.024	0.36316	(0.29000, 0.43395)
0.032	0.36316	(0.29999, 0.44945)
0.054	0.47018	(0.38328, 0.52753)
0.077	0.47018	(0.40919, 0.53437)
0.100	0.47018	(0.41407, 0.54466)
0.208	0.51316	(0.44473, 0.58318)
0.316	0.51316	(0.44844, 0.59074)
0.554	0.68421	(0.59829, 0.74396)
0.772	0.68421	(0.61667, 0.75433)
1.000	0.71579	(0.64233, 0.79893)
2.081	0.83947	(0.77870, 0.88693)
3.162	0.83947	(0.78407, 0.89009)
5.442	0.91053	(0.84390, 0.94347)
7.721	0.91842	(0.87462, 0.95088)
10.000	0.91842	(0.88048, 0.95665)
20.811	0.95789	(0.90874, 0.97336)
31.622	0.95789	(0.92703, 0.97471)
100.000	0.95789	(0.93254, 0.97540)
316.220	0.95789	(0.93469, 0.97766)
1000.000	0.96316	(0.93873, 0.98522)

Table 3. - Isotonic regression of fault latency

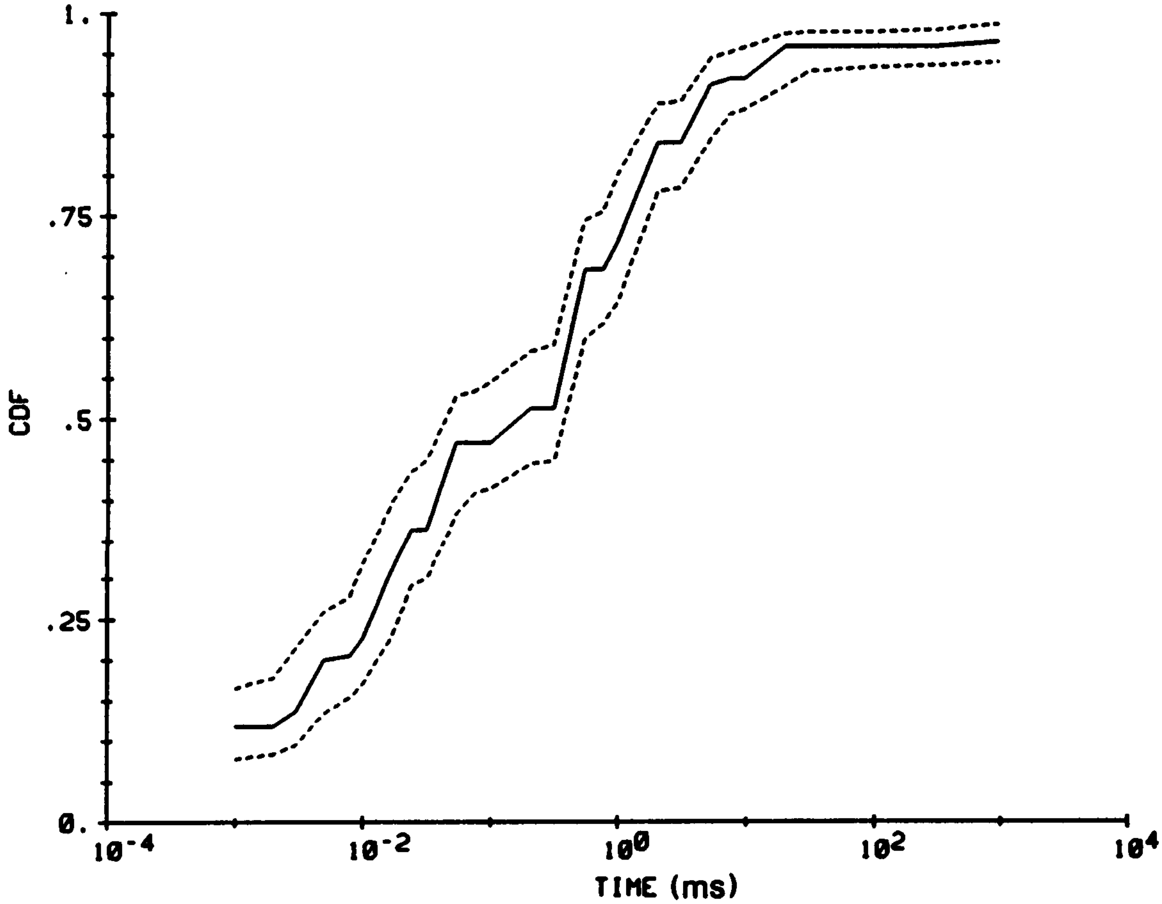

Figure 3. - Isotonic regression of fault latency

Figure 4 shows the Shin estimate of the distribution and the standard binomial model confidence bounds.

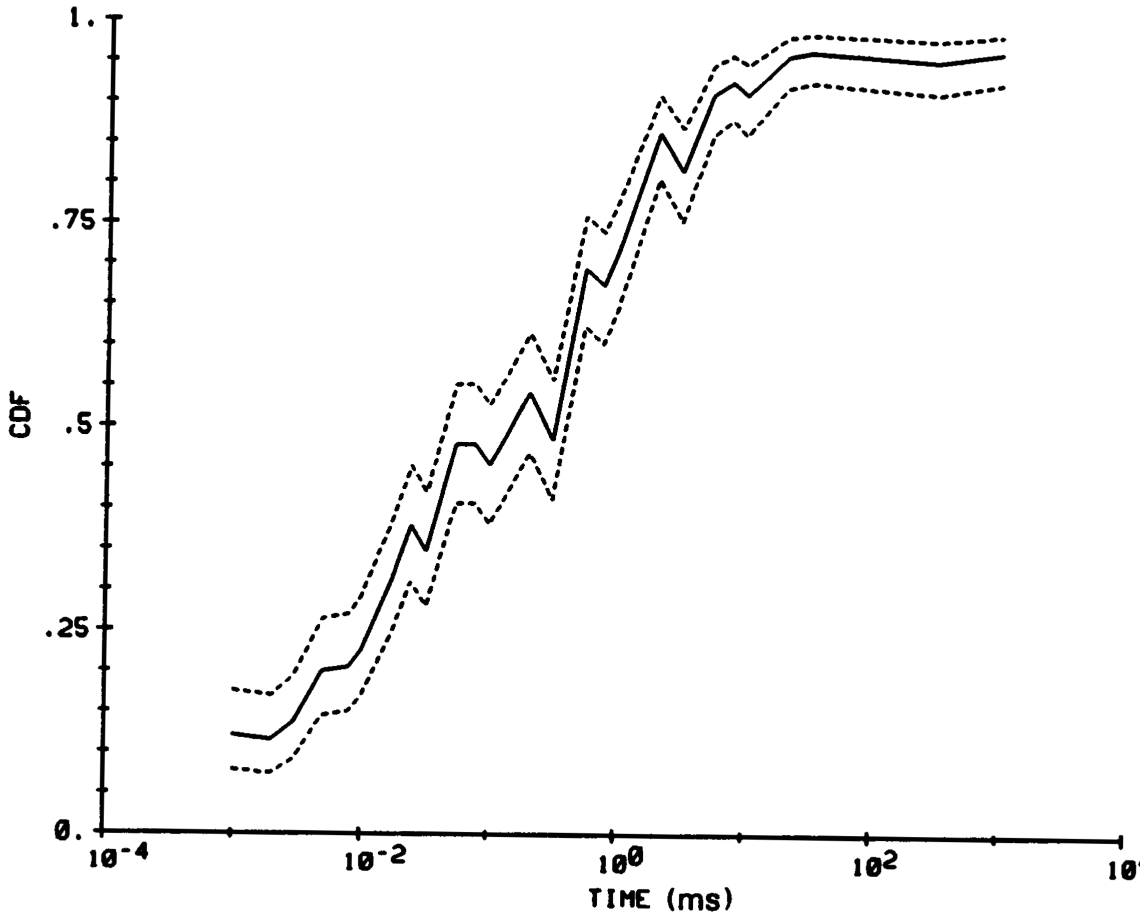

Figure 4. - Shin estimate and binomial confidence bounds

An additional analysis was performed to determine whether fault latency is different in different areas of the processor. In the sample, 13 of the injection locations were in the memory of the processor, 5 in the CPU and 1 in the broadcast register. An isotonic regression analysis was performed on each of these three areas separately. The results are shown in figure 5.

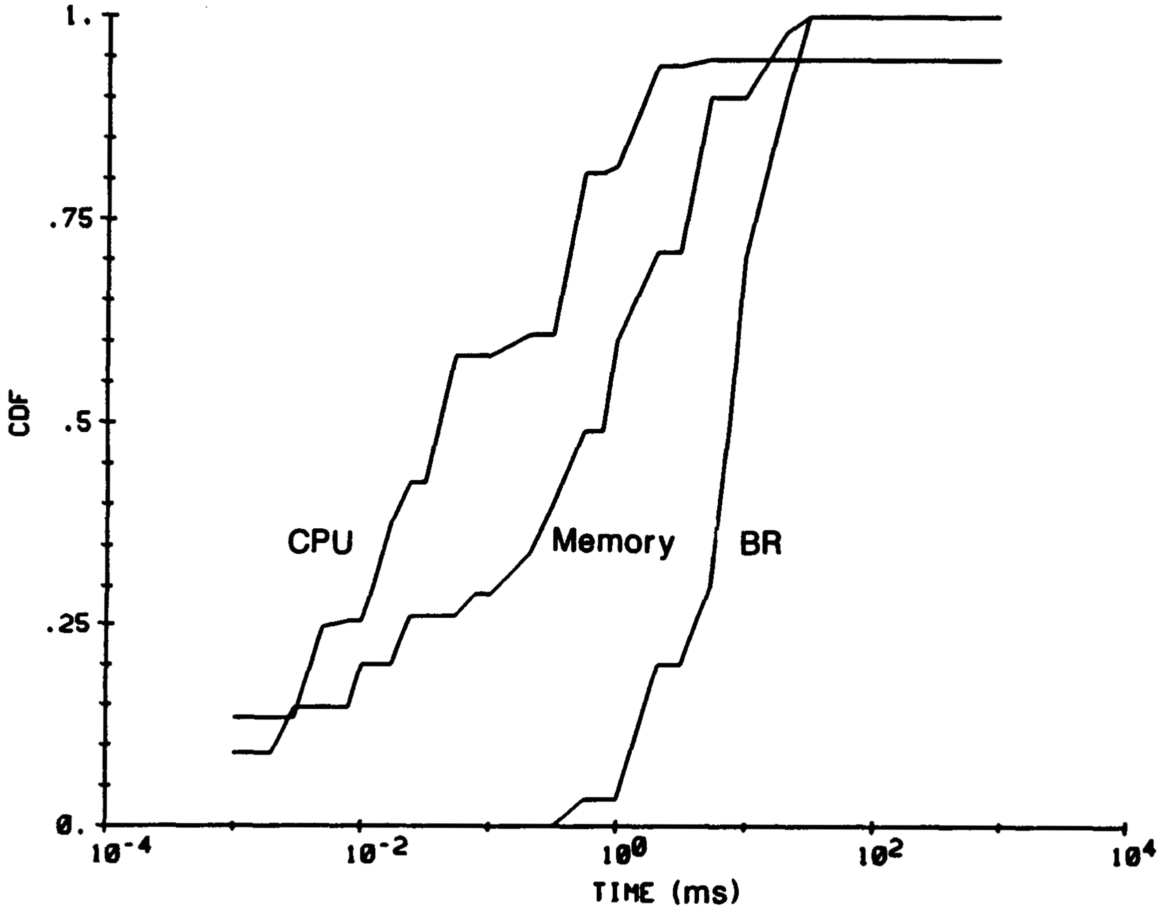

Figure 5. - Fault latency in CPU, memory, and broadcast register

It can be seen that that the average fault latency is greater in memory than on the CPU in this processor. Also, the variance of fault latency is greater in memory than in the CPU. Since only 10 injections were performed for each t_i in the broadcast register, the mean and variance of fault latency can not be characterized with appreciable confidence.

PARAMETRIC DISTRIBUTION FITTING

It is useful in some instances to represent the distribution of fault latency in a parametric form. If $P(t;\underline{\mu})$ is the distribution function for the parametric distribution of interest, where $\underline{\mu}$ is the vector of parameters, then $P(t;\underline{\mu}_0)$ is said to be the 'best fit' if $\underline{\mu} = \underline{\mu}_0$ minimizes the expression

$$\Sigma_{i=1}^{r} \, [F^*(t_i) - P(t_i;\underline{\mu})]^2 \qquad (16)$$

where $F^*(t_1),\ldots,F^*(t_r)$ are the isotonic regression estimates. The downhill simplex algorithm was used to minimize (16) for the two parameter distributions (i.e., Normal, Gamma, and Weibull) [Press, 1986].

The data from table 3 was compared to the following parametric forms:

Exponential	$P(t;\underline{\mu}) = 1-e^{-\lambda t}$
Normal	$P(t;\underline{\mu}) = \int_{-\infty}^{t} (2\pi\sigma^2)^{-1/2} \exp[(x-\mu)^2/2\sigma^2]\, dx$
Gamma	$P(t;\underline{\mu}) = \int_0^t \lambda^\alpha/\Gamma(\alpha) \exp(-\lambda x)\, x^{\alpha-1}\, dx$
Weibull	$P(t;\underline{\mu}) = 1-\exp[(-t/\lambda)^{1/\alpha}]$

The parameters of the best fit for four parametric distributions are given in table 4. The distribution function, the value of (16) at $\underline{\mu}_0$, the $\underline{\mu}_0$, and the mean and variance of $P(\cdot;\underline{\mu}_0)$ are also reported. From this table, it is clear that the Weibull distribution provides the best fit of the four forms considered.

Distribution	Squared Error Loss	$\underline{\mu}_0$	mean	variance
Exponential	0.7788	$\lambda_0 = 4.60$	0.217	0.0472
Normal	0.2836	$\mu_0 = 0.374$ $\sigma_0^2 = 0.444$	0.374	0.444
Gamma	0.0489	$\lambda_0 = 0.170$ $\alpha_0 = 0.224$	1.32	7.76
Weibull	0.0296	$\theta_0 = 1.30$ $\alpha_0 = 0.325$	2.95	183

Table 4. - Results of distribution fitting (ms)

The exponential distribution provided the worst fit. This strongly suggests that in a reliability analysis which includes fault latency, a pure Markov model should not be used. The effect of the non-exponential shape of the distribution should be investigated with the more general semi-Markov model.

CONCLUSIONS

In this paper an indirect statistical method is presented for estimating the distribution of fault latency in a digital processor. The method depends upon the availability of a fault-injector where the duration of an injected fault can be controlled. The method also requires a 100% error detection mechanism that is usually available in a fault-tolerant systems. An experiment was conducted on the SIFT computer system to illustrate the feasibility of the method. The fault latency was found to vary significantly over different locations in the processor. Four distributions were fit to the data. Of the four distributions analyzed, the Weibull distribution was found to give the best fit and the exponential distribution the worst. Finally, a method was presented to calculate confidence bounds for the estimated distribution.

APPENDIX A

FAULT INJECTION DATA

In this section the raw data from the fault-injection experiment is given. The first column contains the fault injection durations. The other columns contain the results of the injections at a particular location. For example the second column contains the fraction of detections (D_i/n_i) for the CPU chip U38. The specific pin injected on the chip can be found by examining table 1. There were 10 injections performed for each injection-duration for each pin (i.e. $n_i = 10$ for all i).

INJ TIME	CPU/U38	CPU/U35	CPU/U29	CPU/U04	CPU/U70
0.001	0.00	0.00	0.00	0.10	0.40
0.002	0.10	0.10	0.00	0.00	0.20
0.003	0.00	0.30	0.00	0.00	0.70
0.005	0.10	0.30	0.00	0.00	0.20
0.008	0.00	0.10	0.00	0.10	0.40
0.010	0.10	0.20	0.00	0.10	0.60
0.017	0.10	0.20	0.00	0.00	0.70
0.024	0.00	0.50	0.20	0.20	0.60
0.032	0.10	0.30	0.10	0.10	0.60
0.054	0.00	0.40	0.00	0.10	0.70
0.077	0.10	0.60	0.20	0.10	0.60
0.100	0.10	0.40	0.00	0.00	0.80
0.208	0.20	0.50	0.40	0.00	0.60
0.316	0.20	0.90	0.30	0.00	0.60
0.554	0.50	0.70	0.30	0.10	0.90
0.772	0.40	0.70	0.40	0.20	0.70
1.000	0.50	1.00	0.50	0.00	1.00
2.081	0.90	1.00	0.60	0.40	0.90
3.162	0.80	1.00	0.40	0.10	1.00
5.442	1.00	1.00	1.00	0.50	1.00
7.721	1.00	1.00	1.00	0.60	1.00
10.000	1.00	1.00	1.00	0.40	1.00
20.811	1.00	1.00	1.00	0.90	1.00
31.622	1.00	1.00	1.00	1.00	1.00
100.000	1.00	1.00	1.00	1.00	1.00
316.220	1.00	1.00	1.00	1.00	1.00
1000.000	1.00	1.00	1.00	1.00	1.00

INJ TIME	BR/U81	MPM1/U21	MPM1/U12	MPM1/U45	MPM1/U59
0.001	0.00	0.00	0.20	0.30	0.00
0.002	0.00	0.00	0.00	0.00	0.00
0.003	0.00	0.30	0.00	0.30	0.10
0.005	0.00	0.20	0.10	0.10	0.00
0.008	0.00	0.00	0.00	0.20	0.20
0.010	0.00	0.20	0.40	0.60	0.10
0.017	0.00	0.20	0.30	0.10	0.10
0.024	0.00	0.20	0.10	0.10	0.20
0.032	0.00	0.20	1.00	0.60	0.20
0.054	0.00	0.10	0.40	0.10	0.50
0.077	0.00	0.10	0.70	0.10	0.40
0.100	0.00	0.50	1.00	0.50	0.30
0.208	0.00	0.60	0.50	0.20	0.50
0.316	0.00	0.60	0.90	0.80	0.20
0.554	0.10	0.50	1.00	0.40	0.80
0.772	0.00	0.80	0.50	0.40	0.80
1.000	0.00	0.80	0.90	0.90	1.00
2.081	0.20	1.00	1.00	0.90	1.00
3.162	0.20	0.70	1.00	1.00	1.00
5.442	0.30	1.00	1.00	1.00	1.00
7.721	0.50	1.00	1.00	1.00	1.00
10.000	0.70	0.70	1.00	1.00	1.00
20.811	0.90	1.00	1.00	1.00	1.00
31.622	1.00	0.80	1.00	1.00	1.00
100.000	1.00	0.70	1.00	1.00	1.00
316.220	1.00	0.60	1.00	1.00	1.00
1000.000	1.00	0.80	1.00	1.00	1.00

INJ TIME	MPM2/U21	MPM2/U11	MPM2/U17	MPM2/U41	MPM2/U52
0.001	0.10	0.00	0.20	0.10	0.20
0.002	0.00	0.20	0.20	0.10	0.20
0.003	0.10	0.00	0.10	0.10	0.20
0.005	0.30	0.00	0.20	0.30	0.20
0.008	0.40	0.20	0.20	0.10	0.20
0.010	0.10	0.10	0.30	0.10	0.30
0.017	0.40	0.40	0.30	0.10	0.50
0.024	0.60	0.60	0.60	0.10	0.40
0.032	0.10	0.00	0.70	0.30	0.60
0.054	0.90	0.70	0.80	0.50	0.50
0.077	0.70	0.80	0.50	0.30	0.30
0.100	0.30	0.10	0.70	0.50	0.70
0.208	0.80	0.70	0.80	0.50	0.60
0.316	0.20	0.30	0.80	0.40	0.40
0.554	1.00	1.00	0.90	0.50	0.80
0.772	0.90	1.00	0.90	0.40	1.00
1.000	0.30	0.70	1.00	0.50	1.00
2.081	1.00	1.00	1.00	0.50	1.00
3.162	0.90	1.00	1.00	0.50	1.00
5.442	1.00	1.00	1.00	0.50	1.00
7.721	1.00	1.00	1.00	0.50	1.00
10.000	1.00	1.00	1.00	0.50	1.00
20.811	1.00	1.00	1.00	0.50	1.00
31.622	1.00	1.00	1.00	0.50	1.00
100.000	1.00	1.00	1.00	0.50	1.00
316.220	1.00	1.00	1.00	0.50	1.00
1000.000	1.00	1.00	1.00	0.50	1.00

INJ TIME	MPM2/U37	MPM2/U35	MPM2/U28	MPM2/U53C
0.001	0.10	0.50	0.00	0.10
0.002	0.10	0.40	0.30	0.30
0.003	0.00	0.20	0.00	0.20
0.005	0.20	0.90	0.60	0.10
0.008	0.30	0.60	0.50	0.40
0.010	0.10	0.80	0.00	0.20
0.017	0.50	0.80	0.80	0.40
0.024	0.40	0.90	0.90	0.60
0.032	0.00	1.00	0.30	0.40
0.054	0.90	0.90	0.90	0.70
0.077	0.80	1.00	0.90	0.90
0.100	0.60	1.00	0.30	0.80
0.208	0.80	1.00	1.00	0.60
0.316	0.30	1.00	0.40	0.90
0.554	1.00	1.00	0.80	0.90
0.772	0.90	0.90	1.00	0.90
1.000	0.80	1.00	0.70	1.00
2.081	1.00	1.00	1.00	1.00
3.162	1.00	1.00	0.90	1.00
5.442	1.00	1.00	1.00	1.00
7.721	1.00	1.00	1.00	1.00
10.000	1.00	1.00	1.00	1.00
20.811	0.90	1.00	1.00	1.00
31.622	1.00	1.00	1.00	1.00
100.000	1.00	1.00	1.00	1.00
316.220	1.00	1.00	1.00	1.00
1000.000	1.00	1.00	1.00	1.00

REFERENCES

1. J. G. McGough and F. L. Swern; Measurement of Fault Latency in a Digital Avionic Mini Processor, NASA Contractor Report 3462, Oct. 1981.

2. J. G. McGough and F. L. Swern; Measurement of Fault Latency in a Digital Avionic Mini Processor, NASA Contractor Report 3651, Jan. 1983.

3. R. Chillarege and R. K. Iyer; Fault Latency in the Memory - AN Experimental Study on VAX 11/780, Proceedings of the 16th Annual International Symposium on Fault-Tolerant Computing, pp. 258-263, 1986.

4. K.G. Shin and Y. H. Lee; Measurement and Application of Fault Latency, IEEE Transactions on Computers, Vol. C-35, No. 4, pp. 370-375, April 1986.

5. Geist, Robert M; and Trivedi, Kishor S.: Ultrahigh Reliability Prediction in Fault-Tolerant Computer Systems. IEEE Trans. Comput., C-32, no. 12, Dec. 1983, pp. 1118-1127.

6. Bavuso, S. J.; and Petersen, P. L.: CARE III Model Overview and User's Guide (First Revision). NASA TM-86404, 1985.

7. R.E. Barlow, D.J. Bartholomew, J.M. Bremner, and H.D. Brunk: Statistical Inference under Order Restrictions, John Wiley & Sons, New York, 1972.

8. Schoenfeld, David A.: Confidence Bounds for Normal Means Under Order Restrictions, With Application to Dose-Response Curves, Toxicology Experiments, and Low-Dose Extrapolation, Journal of the American Statistical Association, Vol. 81, No. 393, pp. 186-195, March 1986.

9. Goldberg, Jack; Kautz, William H.; Melliar-Smith, P. Michael; Green, Milton W.; Levitt, Karl N.; Schwartz, Richard L.; and Weinstock, Charles B.: Development and Analysis of the Software Implemented Fault Tolerance (SIFT) Computer. NASA CR-172146, 1984.

10. Press, Flannery, et. al.: Numerical Recipes, Cambridge University Press, Cambridge, England, 1986, pp. 289-293, 739-740.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM-100521		

4. Title and Subtitle	5. Report Date
Estimating the Distribution of Fault Latency in a Digital Processor	November 1987
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
Erik L. Ellis and Ricky W. Butler	
	10. Work Unit No.
	505-66-21-01
9. Performing Organization Name and Address	11. Contract or Grant No.
NASA Langley Research Center Hampton, VA 23665-5225	
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Technical Memorandum
National Aeronautics and Space Administration Washington, DC 20546-0001	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

This paper presents a statistical approach to measuring fault latency in a digital processor. The method relies on the use of physical fault injection where the duration of the fault injection can be controlled. Although a specific fault's latency period is never directly measured, the method indirectly determines the distribution of fault latency.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Fault latency, reliability, fault tolerance, computer systems	Unclassified - Unlimited Subject Category 62

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	22	A02

NASA FORM 1626 OCT 86